



TSO Introduction

CNS Document ID: D0037

Last Updated: 06/20/2002

This document provides a brief introduction to TSO at CNS. It includes how to sign on and off, how to change your password, how to submit a batch job, descriptions of some of the services available under TSO, and where to obtain more detailed information.

UF Computing & Networking Services

112 Bryant Space Sciences Bldg, University of Florida

P.O. Box 112050

Gainesville Florida 32611-2050

(352) 392.2061

<editor@cns.ufl.edu>

Table of Contents

Introduction	3
About TSO	3
Signing On and Off	4
Working Environment	7
TSO HELP Facility	7
The z/OS (OS/390) File System	8
Custom Command Procedures	11
Application Programs, Languages and Utilities	12
Submitting Batch Jobs	12
Routing Jobs to the Fetch Queue	12
Saving Output to a Disk Data Set	13
Cancelling Jobs from TSO	13
Additional TSO Sign-On Options	14
<i>Size</i> Parameter	15
Reconnect Option	15
Additional Information	16

Introduction

IBM's Time-Sharing Option (TSO) can be used for the creation, modification, and retrieval of z/OS (OS/390) data sets; the interactive execution of programs; and for submitting and viewing batch jobs. TSO includes a full-screen editor (ISPF), a JES2 interface for the submission of z/OS (OS/390) batch jobs, an extensive HELP system, and a service for viewing batch job output (IOF).

CNS's TSO service is named NERTSO. Use of the NERTSO service incurs charges which are billed to your userid. See the *CNS Charging Algorithm* (CNS document D0001) [<http://docweb.cns.ufl.edu/docs/d0001/>] for TSO charges.

This document provides a brief introduction to TSO at CNS. It refers to other sources for more detailed information. It is intended for new TSO users, as a general introduction and overview, and for experienced TSO users, to provide an orientation to CNS's implementation of TSO, and local specifics.

About TSO

IBM's Time-Sharing Option (TSO) may be new to you, but it is actually not new at all; in fact, it is one of the older "operating systems" around. Strictly speaking, TSO is *not* an operating system; rather, z/OS (formerly known as OS/390) is the operating system, and TSO is the *shell*, or user interface, by which users interact with the operating system proper. z/OS (OS/390) and TSO have their origins back in the days when computers were uncommon, and very expensive. In those days, computer time was such a scarce resource, that everything about programming and using computers was geared to make things as easy as possible for the *computer*. You wouldn't want to waste valuable (and expensive) computer time on a task (such as determining how much disk space is needed to save a file), if a human could do it instead; human time was much cheaper and more abundant than computer time. So all aspects of interaction with the computer were arranged such that the users would provide the computer with as easy a task as possible.

This is in clear contrast with the philosophy of modern, "consumer-oriented" computer systems, such as Windows, MacOS, etc., which strive to be "user-friendly," and insulate the user from the "nuts-and-bolts" of the system, insofar as possible. Where Windows says, "I'll handle the details for you," TSO says, "You must explicitly specify, precisely and in complete detail, exactly what you want done."

So why use TSO? Because it is your gateway to the power of CNS's large-scale IBM processor complex, giving you access to a degree of speed and power of processing which far outstrips the ability of even the fastest desktop computer. In addition, jobs running under TSO may (if granted appropriate access permission) access data sets (files) residing on CNS's file system which would not be accessible to a program running on your PC.

While you do have to learn a few things about TSO in order to *use* TSO, keep in mind that much of your time in TSO will be spent working with "helper programs" provided to make it easier for you to accomplish your work. In particular, you will probably spend most of your TSO time inside ISPF/PDF and IOF, two programs which provide dialog-box prompts and menus to assist you with most TSO tasks. Those programs are discussed in other CNS documents which you will learn about later.

Signing On and Off

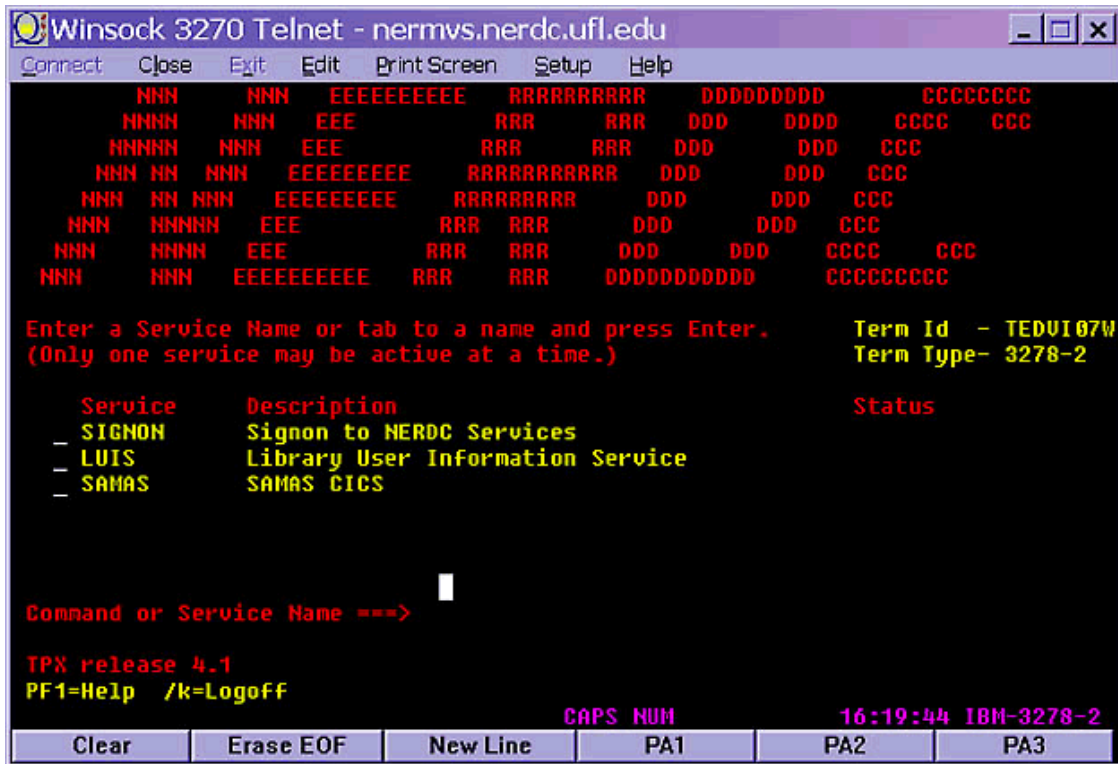
Signing On

Most users will be working from PCs (or Macintoshes, UNIX workstations, or similar general-purpose microcomputers) connected to the Internet. Using a TN3270 emulator program (such as QWS3270, Hummingbird Host Explorer, or Comet) connect to **nermvs.cns.ufl.edu**.

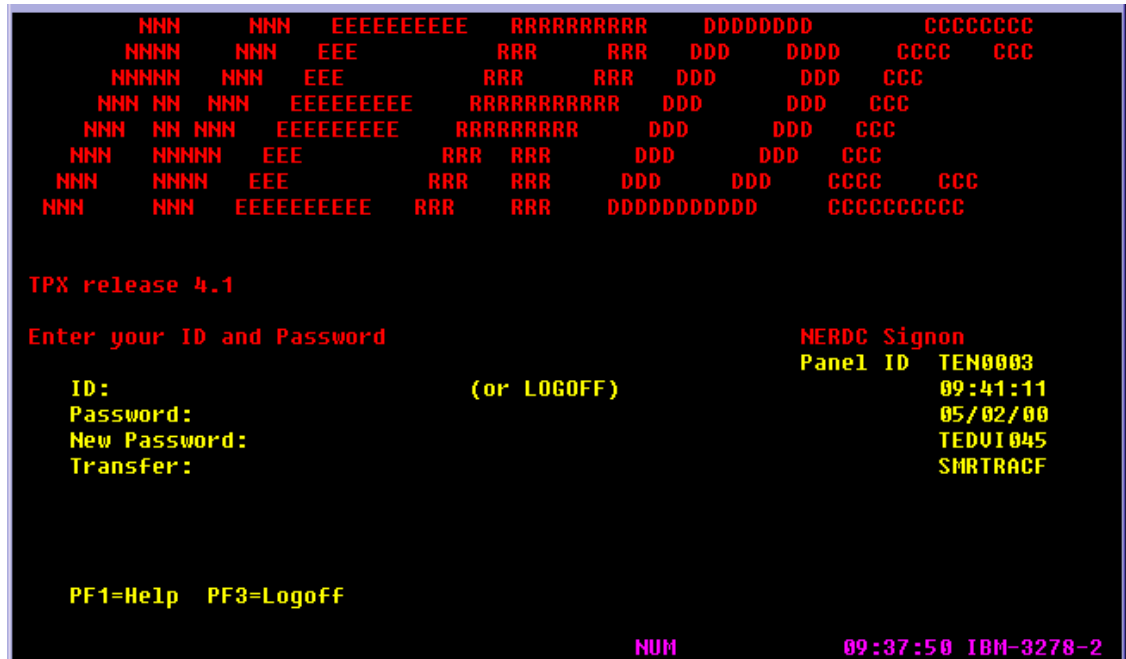
If you have an actual hardware 3270-type terminal connected to an IBM 3745 communications controller on the UF campus network, then when you turn on your terminal, you should see "NERDC VTAM IS ACTIVE". To sign on to TSO from the NERDC VTAM IS ACTIVE screen, type the following:

menu (press <Enter>)

In either case (whether you are using a PC with TN3270, or you type "menu" at the NERDC VTAM IS ACTIVE prompt on a 3270-type terminal), you will see the following screen.



From this screen, either type **signon**, or press your <Tab> key once (so that your cursor is next to the _ SIGNON selection), and press <Enter>. You will then see the following screen.



Enter your userid and password (your password will not be displayed).

Changing Your Password

If you wish, you may change your password on the screen shown above, by entering a new password in the **New Password:** field. After you press **<Enter>**, the system will prompt you to **PLEASE REENTER YOUR NEW PASSWORD FOR VERIFICATION**. Type your new password again, and press **<Enter>**. The next time you use this userid, you will need to use your new password. Note that this changes your password for CNS's NERCICS and dial-up services, in addition to the TSO service. However, it does NOT change your password for NERSP or GatorLink.

Note

Users considering using the UF/CNS dial-up service should be aware that this service is under review, and may possibly be discontinued at or shortly after the end of calendar year 2006. For more information, please see Dr. Hoit's memo to Deans, Directors and Department Heads of 05/02/2006, titled *Charging for UF Dialup Services* [<http://www.admin.ufl.edu/ddd/default.asp?doc=11.11.1920.1>].

Entering TSO

After you have signed on, you will see the NERDC Interactive Services Menu. Your menu may have more or fewer selections than this, but it should generally resemble the one shown below.

```
NERDC Interactive Services Menu for USERID Panel Id - TEN0041
Enter a Service Name or tab to a name and press Enter. Term Type- 3278-2
Multiple services can be active at once. Term Id - TEDUI11Q

Service      Description      Status
- ADMIN      UF Admin. Applications (NERCICS)
- ADMIN2     Second Admin. Session (NERCICS)
- SAMAS      SAMAS
- LUIS       Library User Information Service
- UM         UM/CMS (NERUM)
- TSO        TSO System
- DOCSYS     NERDC Documentation System
- UFINFO     Campus-wide Information Service

TPX release 4.1

Command or Service Name ==>
You are signed on to TPX. Remember to logoff when you are finished.
PF1=Help PA1=This Menu NONE=Jump NONE=Print /k=Logoff
NUM 12:01:59 IBM-3278-2
```

To enter TSO, either type the command **tso** at the Command or Service Name ==> prompt at the bottom of the screen, or move your cursor to the **TSO** line on the menu. Then press **<Enter>**. You should then see the actual TSO presentation screen, including the TSO system **READY** prompt, as illustrated below.

```
ICH70001I USERID LAST ACCESS AT 12:05:09 ON TUESDAY, MAY 2, 2000
IKJ56455I USERID LOGON IN PROGRESS AT 12:17:44 ON MAY 2, 2000
IKJ56951I NO BROADCAST MESSAGES
READY

NUM 12:20:20 IBM-3278-2
```

Signing Off

The command to sign off a TSO session is

logoff

Working Environment

While you do have to learn a few things about TSO in order to *use* TSO, keep in mind that much of your time in TSO will be spent working with "helper programs" provided to make it easier for you to accomplish your work. In particular, you will probably spend most of your TSO time inside ISPF/PDF and IOF, two programs which provide dialog-box prompts and menus to assist you with most TSO tasks.

Most of your work in TSO will be done using the following products:

ISPF

The Interactive System Productivity Facility (ISPF) and the Program Development Facility (ISPF/PDF) are available under TSO at CNS. ISPF and ISPF/PDF provide a full-screen editor and data set manager which offers a menu-driven environment for working with z/OS data sets. Online documentation for ISPF can be found from IBM at <http://www-4.ibm.com/software/ad/ispf/library/> and <http://www-4.ibm.com/software/ad/ispf/support/>. [<http://www-4.ibm.com/software/ad/ispf/support/>]

In addition, CNS has two documents which give an overview of ISPF, and basic instructions for a few tasks such as creating and editing simple files, and submitting batch jobs from ISPF. See D0040, *ISPF at CNS* [<http://docweb.cns.ufl.edu/docs/d0040/>], and D0089, *ISPF: Introduction to the ISPF Editor* [<http://docweb.cns.ufl.edu/docs/d0089/>].

IOF

IOF is a program through which you can monitor and control the processing of your jobs before they are printed. Softcopy (Acrobat/PDF format) documentation can be found on-line from Triangle Systems, the publisher of IOF, at <http://www.triangle-systems.com/iofdoc.shtml>.

In addition, CNS has a document which gives an overview of IOF, including basic instructions for a few tasks such as reviewing job status, cancelling, printing, and saving output. See D0030, *IOF; The Interactive Output Facility* [<http://docweb.cns.ufl.edu/docs/d0030/>].

Several additional TSO commands are available that were written locally or obtained from the SHARE [<http://www.share.org>] library. Type **HELP NERDC** to get a list of locally installed, non-IBM commands.

TSO HELP Facility

The TSO HELP facility provides help for all supported IBM and non-IBM TSO commands. HELP is available while in **READY** mode and from within various TSO commands. A response to a HELP request can vary depending on which mode you are using when you enter the request.

During a TSO session, you can get a partial list and description of IBM TSO commands by entering the word **HELP**. Not all the IBM commands listed are supported locally, and several additional TSO commands are available that were written locally or obtained from the SHARE library. Type **HELP NERDC** to get a list of locally installed, non-IBM commands. Type **HELP HELP** for additional information on using the HELP command.

The z/OS (OS/390) File System

In order to successfully work in TSO, you need to have a basic understanding of the z/OS (OS/390) file system. By convention, under z/OS (OS/390) and TSO, "files" are referred to as "data sets."

If you are familiar with Windows, MS-DOS, MacOS, or even UNIX (e.g. the NERSP operating system), then you are acquainted with the notion of an "hierarchical file system." An hierarchical file system uses the concept of subdirectories (often called "folders") and allows you to create files within subdirectories within subdirectories, pretty much without restriction. z/OS (OS/390) uses a "flat" file system, as opposed to an "hierarchical file system." This means that all z/OS (OS/390) data sets exist at the same "level"--there are no "subdirectories." There are ways to group data sets, based on naming schemes, but these are not really "folders" or "directories," and do not generally behave in an hierarchical fashion. However, that said, sometimes you will perform some tasks in TSO "as if" there were an hierarchical directory structure. Just keep in mind that, in actuality, TSO has no "subdirectories" or "folders" as we commonly think of such things.

Data Set Names

In general, an z/OS (OS/390) data set name will be of the form:

something1.something2.something3

or

something1.something2.something3(whichthing)

In the above example, each of the "somethings" is referred to as a "qualifier." There may be more qualifiers than this, but there will rarely be fewer than 3. Sometimes, and for some purposes, you may think of each of the qualifiers as being like a "directory level" (with the final qualifier being thought of as the "file name"); then the "periods" separating the qualifiers become analogous to the "backslash" character in a Windows file-name. However, as mentioned above, it is important to bear in mind that the true nature of the z/OS (OS/390) file system is NOT hierarchical, and these qualifier levels are not truly "directories."

Each qualifier may be from one (1) to eight (8) characters in length, and the entire name, including the "periods" used to separate the qualifiers, may be a maximum of 44 characters long. The qualifiers may be made up of any alphabetic characters, any numbers, and the characters @, #, and \$. The first character of each qualifier must be an alphabetic character (A-through-Z). z/OS (OS/390) data set names are NOT case sensitive, so "a.data.set.name" is the same as "A.DATA.SET.NAME". The system will convert any data set name you type to all upper-case.

Top-Level Qualifier

The first qualifier ("something1" in the above examples) must be one of a small set of approved "top-level qualifiers" defined by CNS. For most data sets, the top-level qualifier ("something1") will be either "U" or "UF". Some offices/departments on campus have made arrangements with CNS to define special top-level qualifiers for their use; your supervisor or local computer support desk should be able to tell you if your office has a special top-level qualifier which you should use. Any data set lacking a valid (defined by CNS) top-level qualifier will be deleted overnight.

Second-Level Qualifier

The second qualifier ("something2" in the above examples) should generally be your CNS userid. You may encounter some data set names where this qualifier is a letter followed by several numbers (e.g. A0001234). These data set names are using the CNS *account number* in place of the userid. Your local computer support desk can tell you if your office uses the account number convention for its data sets; however most users will have their userids as the second level qualifier for all their data set names. The userid information is required in the second level for accounting purposes, and any data set lacking valid userid information in the second level qualifier cannot be *cataloged*, and therefore will be archived to tape, and deleted from the disk pack overnight. See the section on "Cataloging Data Sets," below, for more information.

Third-Level Qualifier, etc.

The third qualifier ("something3" in the above examples) is entirely up to you (or the policy of your office), within the naming restrictions described above. However, bear in mind that you may have more than three levels of qualifiers in your data set name. Your data set could be named, for example,

U.USERID.MYPROJECT

or

U.USERID.CLASSES.ABC1234.EXAMS.FINAL

or

U.USERID.CLASSES.ABC1234.EXAMS.FINAL(F2001)

For more detailed information on CNS's data set naming conventions, see CNS document D0045, *z/OS (OS/390) Disk Data Sets at CNS* [<http://docweb.cns.ufl.edu/docs/d0045/>].

Partitioned Data Sets

You will no doubt have noticed that some of our example data set names (above) have a final "qualifier" in parentheses, at the end of the data set name; e.g. (*whichthing*) in our first example. This indicates that the data set in question is a special kind of file known as a *Partitioned Data Set (PDS)*.

z/OS (OS/390) supports a variety of different types of data set formats, such as Physical Sequential (PS), Partitioned Organization (PO), Virtual Sequential Access Method (VSAM), and others. For the most part, your data sets will be either Physical Sequential (PS) or Partitioned Organization (PO).

A Physical Sequential (PS) data set is a simple, single data set, made up of a collection of records (lines). These are very common, and you may often encounter them as either input to, or output from, your programs. A PS data set is, as the name suggests, a data set consisting of a collection of records which are written sequentially on the storage device.

Partitioned Organization is used to create Partitioned Data Sets (PDSs). A PDS is a single data set which is subdivided into a small (and limited) number of "virtual data sets" called *members*. For many purposes, you can treat each member of a PDS as if it were a data set unto itself. However, you can also deal with the entire collection (all the members of a PDS) as a single data set. Sometimes PDSs are thought of as analogous to a "hanging file folder"--a file folder which can contain (a few) other folders, and which groups them together for easy management.

PDSs are frequently used as a convenience, such as to group together all of your programs. You can sign on to TSO, list the members of your "program library" PDS, and easily select one to edit or run, without having to remember (and type) the full name of each and every one of your programs. The **ISPF** utility makes it very easy to work with PDSs in this fashion.

Creating ("Allocating") Data Sets

A feature of z/OS (OS/390) which is relatively unique to that system is the requirement that all data sets must be *allocated* before they may be used. The process of allocation basically involves declaring the existence of the data set, including its major attributes, such as size, format (PS or PO, etc.), information about how and where it will be stored, and information about the way data will be stored inside it (fixed-length vs. variable-length records, etc.).

You may allocate some data sets using the Utilities option of **ISPF**; others may be allocated by the JCL in programs that you run. Data set allocation can be a complex topic; for a detailed explanation, see the "Creating Data Sets" section of CNS document D0045, *z/OS (OS/390) Disk Data Sets at CNS* [<http://docweb.cns.ufl.edu/docs/d0045/>]. Procedures for creating data sets (both PS and PDS) using **ISPF** are discussed in CNS document D0040, *ISPF at CNS* [<http://docweb.cns.ufl.edu/docs/d0040/>].

Cataloging Data Sets

As a general rule, when you allocate a data set, you will also want to *catalog* it. A cataloged data set name is placed in a table of information, the catalog, so that the system can automatically locate the data set without a specific reference to volume or device type in your JCL. In addition to simplifying your JCL, this also provides easier portability. The data set's location can be changed from tape to disk or vice versa without modification (except for **SETUP** statements).

If your data set is allocated as part of a program job stream, then the same JCL which allocates the data set will generally include a command to catalog the data set.

If you use the **ISPF** Utilities to allocate your data set, then **ISPF** will catalog the data set for you at the same time it allocates the data set, provided that the name you specify is a valid catalog entry.

Uncataloged data sets on CNS-managed volumes will be archived to tape and deleted from the disk pack overnight, regardless of their name.

For more information on cataloging data sets, see the section "Cataloged Data Sets" in CNS document D0045, *z/OS (OS/390) Disk Data Sets at CNS* [<http://docweb.cns.ufl.edu/docs/d0045/>].

Recovering Deleted Data Sets

Users are cautioned to exercise caution in deleting data sets, or creating uncataloged data sets (which will be automatically deleted by the system). CNS has no utility allowing users to recover deleted data sets. While deleted data sets may often be recovered, this requires the assistance of CNS systems staff, and there is a charge for this service. See CNS document D0001, *The CNS Charging Algorithm* [<http://docweb.cns.ufl.edu/docs/d0001/>], for information on current charges for restoring deleted data sets.

Contact the CNS Support Desk [http://www.cns.ufl.edu/info-services/support/support_desk.html] (392-2061, <consult@lists.ufl.edu>) or appropriate CNS systems staff member for assistance if you need to recover a deleted data set.

Custom Command Procedures

TSO allows users to write and execute custom command procedures, combining TSO commands and/or subcommands within a procedural language such as the CLIST or REXX languages, to perform some specific task. Doing so requires a degree of experience and skill in procedural programming, and is not a function which would ordinarily be needed by most users. A detailed explanation of custom command procedure programming is beyond the scope of this document. A brief summary of the available programming interfaces is given below, for the benefit of advanced users who are interested in this topic. If you need further information or assistance in this area, please contact the UF Computing Help Desk [<http://helpdesk.circa.ufl.edu/>] (392-HELP, <<helpdesk@ufl.edu>>) for assistance.

CNS CLIST Support

A CLIST (command list) is a set of TSO commands or subcommands that can be executed to perform some specific task. CLISTs are usually stored as members of a partitioned data set.

Most TSO users use the default CNS CLIST data set (NERDC . CLIST) that is provided when you sign on to TSO. If you want to automatically allocate other CLISTs when you sign on, you must include them in a SYSPROC concatenation. CNS supports only CLISTs that have variable-length blocked records (*RECFM=VB*). If you write your own CLISTs to be included in the SYSPROC concatenation, make sure they are located in a partitioned data set defined with a *VB* record format.

Refer to the IBM manual [http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/LIBRARY] *TSO/E CLISTS* for more information.

NOTE: The CLIST language is no longer actively being developed by IBM, and users are encouraged to write their procedures in REXX instead.

TSO/E REXX

TSO/E REXX is the implementation of the REXX language processor on the z/OS (OS/390) system. It allows you to write command procedures, application front ends, and macros (such as editor subcommands). CNS recommends that you use REXX (instead of the older CLIST language) to write your command procedures, etc. Detailed information on the REXX language is available from IBM [<http://www-306.ibm.com/software/awdtools/rexx/>].

Application Programs, Languages and Utilities

A variety of application programs, languages, and utilities have been installed on NERTSO, and are available to users. Please see the Languages and Programs section of the *CNS Guidebook* [<http://www.cns.ufl.edu/guidebook/>] for a list of currently installed products.

Submitting Batch Jobs

One of the most common tasks you'll perform in TSO is the submission of *batch jobs*. In fact, batch jobs might be the *only* reason you use TSO. A batch job consists of one or more computer programs, possibly including data to be used by those programs, along with control information ("JCL" or "Job Control Language", and "JES2" - "Job Entry Subsystem" commands) which provides essential information to the computer regarding how the job is to be run. Then, the computer schedules the job to run at a time dependent upon factors such as system load, resources required, and the job priority.

This is a very simple process; from the TSO `READY` prompt, you enter the command:

```
submit something.jobname
```

Usually, the "something" in the above example will be your userid; so the most common form of the command will be:

```
submit userid.jobname
```

Often, your program file will be a member of a Partitioned Data Set (PDS); in this case, the general form of the command will most commonly be:

```
submit userid.pdsname(jobname)
```

Commonly, you will first want to look at a job, perhaps make a few modifications, and then submit it. The ISPF Editor allows you to simply type

```
submit
```

from the Editor command prompt to submit the job currently loaded in the editor. See CNS document D0089, *ISPF: Introduction to the ISPF Editor* [<http://docweb.cns.ufl.edu/docs/d0089/>], for more information.

Routing Jobs to the Fetch Queue

Batch jobs can be routed to the TSO *fetch queue* so that the output can be displayed interactively under TSO. The fetch queue is a temporary "holding area" which allows you to view your output on-line, without having to print it. It also improves your turnaround time since you do not have to wait for a job to be printed. If you want to keep the job, you can

choose to print it from the fetch queue. The IOF (Interactive Output Facility) utility available under TSO allows you to view, delete, print, or save-to-file, output routed to the fetch queue.

You can route output to the fetch queue by using the `QUEUE=FETCH` option on the `JOBPARM` statement in your JCL.

```
/*JOBPARM QUEUE=FETCH, INFORM
```

or

```
/*JOBPARM Q=F, I
```

Specifying `FETCH` overrides a `ROUTE` statement in your JCL. If you use the `INFORM` parameter also, you will be notified when the job finishes execution.

Viewing Jobs in the Fetch Queue

Use **IOF** to view any jobs that you have routed to the fetch queue. To invoke IOF, simply enter **IOF** at the `READY` prompt. For more information about the **IOF** command, enter **HELP IOF**. Additional softcopy (Acrobat/PDF format) documentation can be found on-line from Triangle Systems, the publisher of IOF, at <http://www.triangle-systems.com/iofdoc.shtml>.

Printing Jobs in the Fetch Queue

To print a job in the fetch queue using **IOF**, move the cursor to the line containing the job that you want to print and type a "q" (without the quote marks) on the line next to the job. The job then will be routed to the print destination that you specified on the `ROUTE` statement in your JCL.

Fetch Queue Policies

There is no additional charge for having a job sent to, or held in, the fetch queue. All jobs in the fetch queue reside on CNS's SPOOL data sets. Jobs in the fetch queue are purged after three calendar days. However, the SPOOL data sets are a finite resource. If the volume of output in the fetch queue becomes too large and impacts CNS's ability to serve all customers, output in the fetch queue could be purged before three days.

Saving Output to a Disk Data Set

You may use **IOF** to save the output from your program to a disk data set. IOF refers to this function as a "snap." For more information, and detailed instructions, see CNS document D0030, *IOF: The Interactive Output Facility* [<http://docweb.cns.ufl.edu/docs/d0030/>].

Cancelling Jobs from TSO

You can cancel jobs from an interactive workstation by signing on to TSO and typing either **CANCEL** or **PURGE** as shown below. You must be signed on to TSO under the userid from which the job was submitted. The **CANCEL** command requests that JES2 terminate the scheduling or execution of a job. The **PURGE** command requests that JES2 purge a job from its queue. The **PURGE** command will cancel a job and delete any output. The format for the commands is

```
CANCEL J(jobnumber or jobname)
PURGE J(jobnumber or jobname)
```

in which "*jobname*" or "*jobnumber*" is the job to be cancelled.

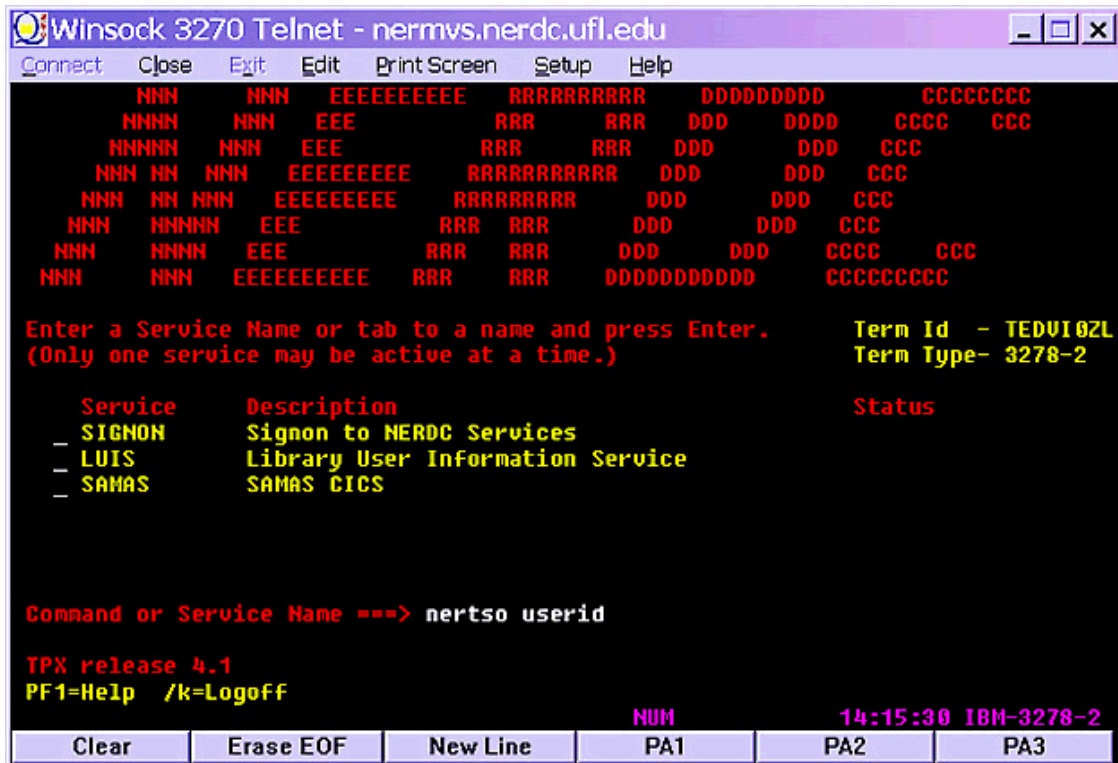
A response of `JOB NOT FOUND` means that the job has already completed or that no job of that name or number is found for that userid; this usually means that the job has already completed.

Jobs can also be cancelled from IOF.

Additional TSO Sign-On Options

Occasionally, you may need to sign on to NERTSO using special options; for example, using a larger memory region, or to reconnect after a broken connection. The following procedure will give you access to the complete TSO logon panel, so that you can specify any additional options needed.

At the regular NERDC Menu Signon screen, instead of selecting `SIGNON`, type **nertso userid** (substitute your actual CNS userid in place of "userid"), as shown in the following image.



You will then see the following panel, which allows you to specify all of the various TSO sign-on options.

```

Winsock 3270 Telnet - nermvs.nerdc.ufl.edu
Connect Close Exit Edit Print Screen Setup Help

----- TSO/E LOGON -----

Enter LOGON parameters below:                                RACF LOGON parameters:

Userid   ==> USERID                                         New Password ==>
Password ==>                                                Group Ident  ==>

Size     ==> 6144

Command  ==> ex 'uf.userid.clist($$$begin)'

Enter an 'S' before each option desired below:
      -Nomail      -Nonotice      S -Reconnect      -OIDcard

PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field
NUM 14:19:53 IBM-3278-2

Clear Erase EOF New Line PA1 PA2 PA3

```

Size Parameter

The *Size* parameter during TSO sign-on allows you to increase the region size. The default region size is 4096K, if *Size* is not specified. In the panel above, the user has requested a *SIZE* of 6144 kilobytes.

Note that CNS charges for memory usage. If you specify a larger region size (and use it), you can expect to receive a slightly higher charge per TSO session.

Reconnect Option

The **Reconnect** option allows you to connect to a TSO session that has been interrupted by a communications failure and continue working from the point of interruption. In the panel above, the user has entered an **S** in the Reconnect field, specifying that this logon is a Reconnect logon.

The Reconnect option works in the following way. When you sign on to TSO, your userid and workstation are allocated to a TSO address space which remains active for the length of your session. If the entire system goes down, or if the TSO subsystem goes down, your TSO address space immediately terminates (it ABENDs). When this occurs, your userid is, in effect, logged off. There is no way to re-establish the session between your workstation and TSO in these situations and any work that was in progress at the time of the failure will be lost.

However, if a communications failure occurs between your workstation and the system, your TSO address space remains active. For dial-up connections, accidentally hanging up the phone or picking up on an extension are considered communications failures. Excessive noise or crosstalk on the line (for dial-up or dedicated connections) could also cause a communications

failure.

If a TSO session is interrupted due to a communications failure, your address space will remain active for 30 minutes before terminating. Even though your workstation may lock up, the TSO address space will remain active and you will continue to be charged "connect time." After communications are restored, you will have an opportunity to reconnect the workstation to the same address space to continue the session from where it left off. The active TSO address space will also prevent you from establishing a new session with TSO (logging on without using the Reconnect parameter) during this 30-minute time period.

Using the RECONNECT Option

To re-establish connection with a session that was interrupted, you must:

- Use the same CNS userid that you used to logon to the session originally;
- Use the same type of terminal or workstation that you were using when the failure occurred;
- Logon within 30 minutes of the time when your session was interrupted;
- Specify the Reconnect option when logging back on, by entering an **S** (for Select) in the `Reconnect` field of the TSO/E LOGON panel (shown above).

If the reconnection is successful, you will see the message " IKT00300I LOGON RECONNECT SUCCESSFUL, SESSION ESTABLISHED". You will then be able to continue your session normally from the point at which you were interrupted.

Please note that no other logon parameters (such as *SIZE*) may be used during a logon which uses the *Reconnect* parameter because the parameters specified during the initial logon for the session to which you are reconnecting will still be in effect. If any other logon parameters are specified during a *Reconnect* logon, they will be ignored.

When to Use the *Reconnect* Option

The *RECONNECT* parameter is designed to work only when a TSO session has been interrupted by a break in the communications link which connects your workstation to the TSO subsystem.

If you try to logon to TSO with the same userid within 30 minutes after your session has been interrupted by a communications failure (while the address space is active) without using the *reconnect* option, you will receive an error message stating that your userid is in use and you will not be able to sign on. In this case you should re-attempt your logon using the *RECONNECT* option.

On the other hand, if you do try to use the *Reconnect* option when you do not have an active TSO address space still allocated to your userid (i.e., after you have signed off or after the system has gone down) or if you try to reconnect using a different type of workstation you will also receive an error message and your logon will be rejected.

Additional Information

For additional information on using TSO at CNS, please refer to the following CNS documents:

- D0030, *IOF: The Interactive Output Facility* [<http://docweb.cns.ufl.edu/docs/d0030/>]
- D0040, *ISPF at CNS* [<http://docweb.cns.ufl.edu/docs/d0040/>]
- D0089, *ISPF: Introduction to the ISPF Editor* [<http://docweb.cns.ufl.edu/docs/d0089/>]
- D0045, *z/OS (OS/390) Disk Data Sets at CNS* [<http://docweb.cns.ufl.edu/docs/d0045/>]

IBM also has a TSO Primer on-line, at <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>. Look for "Z/OS Elements and Features Adobe Acrobat PDF" header. Then select the "V1R2" link. Then select the "TSO/E" link. Then select the "TSO/E Primer" link.

Your Comments are Welcome

We welcome your comments and suggestions on this and all CNS documentation. Please send your comments to:

UF Computing & Networking Services
112 Bryant Space Sciences Bldg, University of Florida
P.O. Box 112050
Gainesville Florida 32611-2050
(352) 392.2061
<editor@cns.ufl.edu>

